

AD-A276 224



2

NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS

KNOWLEDGE DISCOVERY
USING
GENETIC PROGRAMMING

by

Lt. Steven Lee Smith
and
Capt. Mohammed Abdul latif Al-Mahmood

December, 1993

Thesis Advisor:

Balasubramaniam Ramesh

Approved for public release; distribution is unlimited.

DTIC
ELECTE
MAR 04 1994
S B D

7478
94-07074



94 3 03 010

**Best
Available
Copy**

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 16 December 1993	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE KNOWLEDGE DISCOVERY USING GENETIC PROGRAMMING		5. FUNDING NUMBERS	
6. AUTHOR(S) Al-Mahmood, Mohammed A. and Smith, Steven L.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School, Monterey CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE *A	
13. ABSTRACT Dramatic growth in database technology has outpaced the ability to analyze the information stored in databases for new knowledge and has created an increasing potential for the loss of undiscovered knowledge. This potential gains for such knowledge discovery are particularly large in the Department of Defense where millions of transactions, from maintenance to medical information, are recorded yearly. Due to the limitations of traditional knowledge discovery methods in analyzing this data, there is a growing need to utilize new knowledge discovery methods to glean knowledge from vast databases. This research compares a new knowledge discovery approach using a genetic program (GP) developed at the Naval Postgraduate School that produces data associations expressed as IF X THEN Y rules. In determining validity of this GP approach, the program is compared to traditional statistical and inductive methods of knowledge discovery. Results of this comparison indicate the viability of using a GP approach in knowledge discovery by three findings. First, the GP discovered interesting patterns from the data set. Second, the GP discovered new relationships not uncovered by the traditional methods. Third, the GP demonstrated a greater ability to focus the knowledge discovery search towards particular relationships, such as producing exact or general rules.			
14. SUBJECT TERMS Genetic Programming, Knowledge Discovery, Datamining, IDIS, Genetic Algorithms, Inductive Learning, Inductive Rules.		15. NUMBER OF PAGES 85	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)

Prescribed by ANSI Std. Z39-18

DTIC COPY NOT SELECTED 2

Approved for public release; distribution is unlimited.

KNOWLEDGE DISCOVERY
USING
GENETIC PROGRAMMING

by

Steven L. Smith
Lieutenant, United States Navy
M.B.A., University of California, Los Angeles, 1978
and
Mohammed A. Al-Mahmood
Captain, Bahrain Defense Force
B.S.E., University of Petroleum And Minerals, Saudi Arabia, 1985

Submitted in partial fulfillment
of the requirements for the degree of

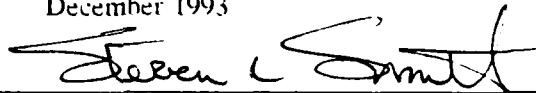
MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the


NAVAL POSTGRADUATE SCHOOL

December 1993

Author:

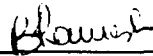


Steven L. Smith

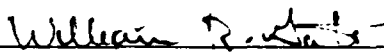


Mohammed A. Al-Mahmood

Approved by:



Balasubramaniam Ramesh, Thesis Advisor



William R. Gates, Associate Advisor



David R. Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

Dramatic growth in database technology has outpaced the ability to analyze the information stored in databases for new knowledge and has created an increasing potential for the loss of undiscovered knowledge. This potential gains for such knowledge discovery are particularly large in the Department of Defense where millions of transactions, from maintenance to medical information, are recorded yearly. Due to the limitations of traditional knowledge discovery methods in analyzing this data, there is a growing need to utilize new knowledge discovery methods to glean knowledge from vast databases.

This research compares a new knowledge discovery approach using a genetic program (GP) developed at the Naval Postgraduate School that produces data associations expressed as IF X THEN Y rules. In determining validity of this GP approach, the program is compared to traditional statistical and inductive methods of knowledge discovery.

Results of this comparison indicate the viability of using a GP approach in knowledge discovery by three findings. First, the GP discovered interesting patterns from the data set. Second, the GP discovered new relationships not uncovered by the traditional methods. Third, the GP demonstrated a greater ability to focus the knowledge discovery search towards particular relationships, such as producing exact or general rules.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

I	INTRODUCTION	1
A.	BACKGROUND	1
B.	OBJECTIVES	3
C.	RESEARCH METHODS	3
D.	THESIS ORGANIZATION	4
II	KNOWLEDGE DISCOVERY	5
A.	BACKGROUND	5
B.	WHAT IS KNOWLEDGE DISCOVERY?	6
1.	Patterns	7
a.	Pattern Representation	9
2.	Certainty	10
3.	Interesting	12
4.	Efficiency	13
III	EVOLUTION OF KNOWLEDGE DISCOVERY SYSTEMS	15
A.	STATISTICAL APPROACHES	15
1.	Statistical Association Methods	16
B.	INDUCTIVE LEARNING	17
1.	CLS and ID3	20
2.	AQ Algorithms	21
3.	IXL/IDIS	23
C.	SUMMARY	24

IV	GENETIC ALGORITHMS / GENETIC PROGRAMMING	26
A.	GENETIC ALGORITHMS	26
B.	SAMUEL: A RULE GENERATING GENETIC ALGORITHM . .	29
C.	GENETIC PROGRAMMING	31
V	NPS GENETIC PROGRAM (NPSGP)	33
A.	BACKGROUND	33
1.	Data Description	33
B.	CONTROL PARAMETERS	35
1.	Fitness Measure	35
2.	Selection Criteria	37
3.	Population and Generation	38
C.	EVOLUTION OF RULES	38
D.	AN EXAMPLE OF THE GP PROCESS	40
E.	GP EVALUATION RUNS	44
F.	SUMMARY	46
VI	COMPARISON OF KNOWLEDGE DISCOVERY RESULTS	47
A.	STATISTICAL ASSOCIATIONS	48
1.	Results:	49
B.	IDIS	51
1.	IDIS Results	53
C.	NPSGP	54
1.	NPSGP Parameters	55
2.	Fitness Functions	56
3.	Fitness Function Results	58

a. Example A:	58
b. Example B:	59
c. Example C:	60
4. Function Set Results	60
a. Example D:	61
D. COMPARISON OF RESULTS	61
VII CONCLUSIONS AND RECOMMENDATIONS	64
A. CONCLUSIONS	64
B. FUTURE RESEARCH	66
1. Fitness Function Applications:	66
2. Different Control Parameters	67
3. Develop Measurements of Rule Fit	68
4. Database Objects	68
C. FUTURE DOD APPLICATIONS	69
APPENDIX A TEST DATA	70
APPENDIX B IDIS PARAMETERS	72
APPENDIX C NPSGP USERS GUIDE	74
LIST OF REFERENCES	76
INITIAL DISTRIBUTION LIST	78

I INTRODUCTION

A. BACKGROUND

In his book *POWERSHIFT*, Alvin Toffler sees the acquisition of "knowledge" as the leverage behind the changing political and social structures in today's technologically oriented world. Industrial knowledge, for example, has been used to gain competitive advantage, allowing increased efficiency and the ability to underprice competitors. Similarly, the cornerstone of the United Nations Coalition success in Desert Storm was the absolute knowledge advantage demonstrated by a technological, tactical and training military superiority over Iraqi military forces.

Historically the acquisition of knowledge has been a time and labor intensive undertaking. But leveraged by computer technology, we now live in an era where the velocity of social, economic and political change has accelerated dramatically. To keep pace and understand these changes, there is a real need to develop new and automated methods of discovering knowledge that circumvent the traditional time consuming methods.

In the Department of Defense, for example, there are vast databases composed of millions of computerized files

recording logistic and maintenance actions. More specifically, at Naval Sea Logistic Center (NAVSEALOGCEN) in Mechanicsburg Pa, the Navy's Material Maintenance Management (3M) system has recorded millions of shipboard maintenance actions in a vast database. These recorded transactions give detailed information regarding maintenance actions; such as the types of equipment and system failures; required parts and actions codes representing steps to correct the problem. Across the street on the same Navy base in Mechanicsburg, the Navy's Ships Parts Control Center (SPCC) has created a vast database of repair parts usage data. This database contains information on shipboard parts usage, including historical demand, manufacturer, quality assurance codes, etc.

These databases offer a potential harvest of new knowledge for future applications of knowledge discovery systems. For example, by using both the 3M and SPCC databases, researchers could develop associations between the types of equipment failures, repair parts usage and mean times between failure to develop predicative associations for the next failure. Instead of waiting for equipment casualties, maintenance personnel could use this information to replace parts in a high risk category of failure.

This thesis reviews different computer aided methods in the acquisition of knowledge. First, it discusses the more traditional methods of developing deductive and inductive

associations between data leading to the discovery of knowledge. After discussing the limitations of these approaches, the thesis demonstrates that newer nontraditional methods of knowledge discovery, that is, genetic programs, can be used successfully as an alternative approach to knowledge discovery.

B. OBJECTIVES

The objective of this research is to determine whether Genetic Programming offers insight into relations among data elements not readily discovered by traditional methods. The ability to develop knowledge is represented by the capability to generate meaningful associations among elements of a database and express these relationships in terms of patterns expressed as rules. This thesis focuses on inductive methods of learning relationships among attributes from a data set and formalizes these relationships into heuristic rules.

The objective is to compare the ability to produce rules from a database using genetic programming with the capability of with traditional statistical and inductive methods.

C. RESEARCH METHODS

The focus of the research is to produce rules from quantitative data using a genetic program developed at the Naval Postgraduate School (NPS). The GP system initially

produces random data associations (rules) and then evolves these rules through an evolutionary adaption and selection process. The same data set was analyzed using traditional statistical methods and a commercial software program using deductive/inductive methods to develop rule associations. The three techniques focuses are compared according to the types of data associations they produce, rather than assessing the value of the information produced.

D. THESIS ORGANIZATION

This document is structured to cover a variety of key concepts prior to comparing the rules generated from the NPS genetic program and other statistical/induction/deduction based methods. First, the general concepts behind knowledge discovery is discussed in Chapter II. The evolution of knowledge discovery systems is covered in Chapter III with a review of the apparent strengths and weaknesses of these traditional systems. Chapter IV discusses the field of genetic algorithms and programs. Chapter V reviews the GP developed at the Naval Postgraduate School for this research and Chapter VI lays out the results of the data analysis of the different approaches. Chapter VII concludes the research by comparing the rules generated and offers suggestions for future applicability involving rule generating genetic programs.

II KNOWLEDGE DISCOVERY

A. BACKGROUND

The recent proliferation of inexpensive computer memory has caused extensive growth in the size and amount of existing data storage capacity. One estimate states the amount of stored information in the world doubles about every two years [Ref. 1:p. 2]. This vast storehouse of unanalyzed data in turn has created a surge in demand for automated methods of shifting through and compressing this raw data into meaningful information, i.e., useful "knowledge".

In addition to the dramatic increases in mass storage capacity of digitized information, there are three other major technologies in the field affecting the direction of information technology and the information infrastructure that supports the foundation for knowledge discovery: on line databases, networking and digitization of information [Ref. 2:p. 18].

On line databases are a major technological advancement playing a role in knowledge discovery. Historically, man's knowledge was archived in the form of books and drawings. Digitizing this information for accessibility offers tremendous advantages over textual information. On line

databases represent a vast and vital storehouse of information that facilitates both the necessary functions of data access and retrieval.

Networking, the connection of computer systems through local and wide area networks, also plays a vital role in developing future knowledge discovery systems. Openness, connectivity, data sharing and interoperability all share the same ability to transfer data to remote sites almost instantly.

Finally, most of the current world's data is not in a digitized medium. The challenge is to translate vast amounts of accumulated knowledge into binary formats for future knowledge discovery. The current method for translating this historical information is through Optical Scanning. Although a relatively new technology, optical scanning represents a potential bridge between the written past and current data storage techniques.

Before going further into the methods of knowledge discovery, the salient characteristics of knowledge will be reviewed to gain insight into the nature of knowledge vice unsubstantiated or meaningless associations.

B. WHAT IS KNOWLEDGE DISCOVERY?

Knowledge discovery is the nontrivial extraction of implicit, previous unknown, and potentially useful information from data. [Ref. 1:p. 3]

If this definition seems somewhat vague it hits at the core definition of knowledge. What is nontrivial to one is vital to another. What is unknown to one may be common knowledge to others. The problem is a lack of a singular defining concept of knowledge to determine if knowledge has actually been discovered. The approach used in this thesis is to envelope the concept of "knowledge" by defining it's attributes. The attributes that are used to surround the concept of knowledge are discussed in detail in the following sections. [Ref. 1:p. 3].

1. Patterns

One way to look at patterns is as a collection or class of records sharing something in common ... Discovering pattern classes is a problem of pattern identification or clustering. [Ref. 1:p. 15]

The first attribute of knowledge and knowledge discovery, is the concept of pattern recognition. Frawley, Piatetsky-Shapiro and Matheus point out two basic approaches to knowledge discovery through pattern recognition: traditional numeric methods and conceptual clustering [Ref. 1:p. 15]. These methods of discovering "patterns" in databases use either statistical correlations or develop heuristic rules, depending on the type of knowledge the researcher is attempting to find. An example of statistical correlations is regression models, where the dependent variable (output) is calculated from a combination of

independent (input) variable weights. Most regression techniques are parametric; they require the user to specify the functional form of the solution. If the underlying form of the function is unknown, parametric methods tend to break down. As an illustration, a sample database of financial information can be analyzed to determine the correct combination of attributes to maximize income, I_a , given constraints, $C_1 \dots C_n$, this problem lends itself to a statistical optimizing approach. The results of this analysis are often in the following form:

$$\text{INCOME} = a \cdot \text{Var}_1 + b \cdot \text{Var}_2 \dots c \cdot \text{Var}_n - d \cdot C_1 \dots - d \cdot C_n$$

(where INCOME is the dependent variable, $a \dots d$ are constants, $\text{Var}_1 \dots \text{Var}_n$ are income independent variables and $C_1 \dots C_n$ are constraint (cost) independent variables.)

Such approaches are termed compensatory. They are based upon the assumption that trade-offs between relevant attributes will maximize (or minimize) the overall evaluation [Ref. 3:p. 217]. There are, however, two basic concerns with this approach, despite evidence of strong predictive performance. First is the concern that compensatory methods are good models for developing data associations, particularly when dealing with non-compensatory associations. The second concern is that numerical coefficients provide limited insight into relationships among the other attributes [Ref. 3:p. 218].

Conceptual clustering, the alternative approach, works with both nominal and structured data. This is a definite advantage over the linear models when analyzing databases. Coupled with logical representations, such as production systems (for example, if X, then Y) clustering offers representations that overcome the restrictive nature of compensatory statistics. For example, using the same financial database to identify inductive relationships results in clustering the data into groups and measuring the validity of these associations by hypothesis testing. From these methods heuristic rules are produced to explain data associations in production contexts:

```

IF
     $a_1 < Var_1 < a_2$ 
AND
     $b_1 < C_2 < b_2$ 
THEN
    INCOME = HIGH

```

(INCOME is the dependent variable, $Var_{1..n}$ and constraints $C_{1..n}$ are the independent variables and $a_{1..2}$, $b_{1..2}$ are constants.)

a. Pattern Representation

Once patterns have been recognized, they must be represented. The intent is to convey statistical associations to a variety of users who may or may not have a basic understanding of statistics. Therefore, these patterns should use standard representations, communicating complex associations. Logical representations are more natural than

statistical representations for computation and can be used in natural language forms. Common logic representations include the production rules, mentioned earlier, relational patterns ($X > Y$) and decision trees (equivalent to ordered lists of rules).

Because a combination of logical formats and natural language is easier to interpret than complex equations, this research focuses primarily on the inductive style of clustering patterns and presenting them in terms of production rules. The use of production system formalism is an important advantage since it is apparently consistent with human reasoning and therefore more easily understood. [Ref. 3:p. 218].

2. Certainty

Rarely is knowledge absolute, particularly when dealing with data containing inexact and missing elements. To overcome noisy and inexact data, knowledge researchers need quantitative measures indicating the level of trust they can place on the knowledge developed by the database discovery programs. Without the ability to attach a subjective level of faith or quantitative measures of confidence, patterns discovered from databases become merely suppositions. Therefore, they never achieve the status of knowledge.

Computing a measure of certainty involves not only the integrity of the data analyzed but also the data sample size. As a means of representing uncertainty, this requires the developing probabilistic information regarding generated rules. One common technique is to augment logical expressions with probabilistic weights indicating probabilities of success. Simple contingency tables can be used either to test predictive validity or lack of statistical association between categorical attributes when a rule agrees or disagrees with the data set. These contingency tables represent the ability of production rules to fit the data set using two components, dependent and independent variables. In terms of a production rules , independent variables are the "input" attributes and values that produce a rule associated with an "output" category of the dependent variable. Expressed in IF X THEN Y format, the dependent variables represent the right hand side (RHS) of the rule while the independent variables represent the left hand side (LHS). For example:

Basic Contingency Table

("Hit" implies the rule conditions were met in the database.)

		(RHS)	
		HIT	NOT HIT
(LHS)	INDEPENDENT	a	b
	VARIABLES	c	d

(Where a are data set and rule category agreements, b and c are errors of omission of either the rule category or data set, and d represents misses to both arguments).

Such contingency tables can measure trust in the rules by several methods. As an example, simple confidence factors, such as $(a / (a + b))$, can be used to indicate the accuracy of the rules in the database. Statistical confidence can be estimated for the parameters of the contingency table using binomial distributions. Contingency tables can also be used to develop frequencies of cross classification, joint and marginal probabilities. [Ref. 4:p. 175]. Binomial standard deviations can also be developed for the conditions in the contingency tables (HIT-HIT, NOT HIT-HIT, etc.).

3. Interesting

Discovered patterns are meaningless if the information contained within these patterns is without relevancy to the user. Patterns must be interesting, that is, useful. "Knowledge is useful when it can achieve a goal of the system or the user." [Ref. 1:p. 4].

To achieve usefulness, knowledge discovery methods must perform functions that filter trivial from non-trivial information based on the user's interest. But such a filtering in itself can be a double edged sword. A knowledge discovery system that incorporates functions for predefining attributes of interest can also *de facto* limit the scope of knowledge discovery. For example, a researcher may be interested in relationships between certain dependent and independent variables, to the exclusion of other variables in the database. Even though a researcher may be searching for patterns about variable X that involve variables Y and Z, the researcher may not know what those patterns look like. They may involve other variables such as A and B. The key to this dilemma is to provide the discovery system the ability to define "areas of interest" and still allow enough autonomy in mining to data for discover unanticipated patterns.

4. Efficiency

There are several efficiency measures for a knowledge discovery system. A major factor in efficiency is the degree of processing time required per unit of data analysis. Searching for knowledge is usually accomplished in large databases requiring significant memory and CPU time. Mitigating this concern are the fact that large knowledge discovery programs, however, are not routine jobs and are

usually done in the background of other computer transactions.

III EVOLUTION OF KNOWLEDGE DISCOVERY SYSTEMS

A. STATISTICAL APPROACHES

Traditionally, relationships in data have been discovered through the statistical methods of deduction and inference. In statistical methods, the quantitative properties of information are categorized into either descriptive or inferential statistics [Ref. 5:p. 2].

Descriptive statistics is a method for organizing and summarizing information. For example, breaking down data into categories and developing percentages represents a descriptive approach (e.g., 1992 election was between three parties; republicans, democrats and independents who received 46, 38 and 12 per cent of the popular vote). Inferential statistics is a method for developing conclusions regarding the data by analyzing a sample of the data. Inferential statistics relies on mathematical models of distribution, e.g., based upon a random sampling of 500 people, CNN newswire stated that 43 per cent of Americans approved of President Clinton's handling of events in Somalia. To effectively classify and generate interesting statements regarding analyzed data, knowledge discovery systems must employ both descriptive and inferential statistics.

1. Statistical Association Methods

There are numerous approaches for developing associations through descriptive and inferential statistics. One is the multivariate linear regression models, as discussed in Chapter II B.1. Another is statistical cluster analysis, a form of conceptual clustering, places objects into groups (clusters) suggested by similarities in the data. Data is summarized in these clusters by similarities in the data characteristics. In cluster analysis, a priori knowledge of the data set is not required.

Discriminate analysis is another approach that selects a subset of the quantitative independent variables that reveal differences among the dependent variable categories [Ref. 6:pp. 39-40]. The selection process is accomplished through statistical association tests, such as the F test, R^2 partial correlation and Wilkes Lambda tests. Discriminate functions are evaluated by estimating the probabilities of misclassifying the data [Ref. 6:pp. 909-910]. However, discriminate analysis, requires a prior knowledge of the categories.

Another statistical analysis method is the process of Reification [Ref. 2:p. 407]. Reification takes a set of database fields (objects) and collapses them into a subset of smaller object fields (called descriptors) that best describe the object's key attributes. Reification techniques include factor analysis, principle component analysis and

cluster analysis. As we shall see later in this chapter, developing the statistical attributes of database objects is a key component to Parsaye's knowledge discovery engines IXL (Induction with eXtremely Large databases) and IDIS (Information Discovery System).

As we have seen, statistical analysis has traditionally laid the foundation for Knowledge Discovery. The ability to classify (descriptive statistics) and formulate conclusions based upon sampling the data (inferential statistics) has been an important first step in knowledge discovery techniques. Further, statistical analysis is also crucial in measuring the quality of discovered knowledge. Parsaye elaborates on the need to extend the statistical approach in two basic ways. First, because statistics is a quantitative approach, output results come in the form of mathematical functions or equations. Since the average user of a database mining system is not a statistician, these complex representations must be converted into formats the user understands. Second, the statistical approach should be transparently built into the discovery system so that the user need not be a statistical expert.

B. INDUCTIVE LEARNING

Inductive learning is defined as the ability to describe a class from a review or analysis of the individual objects

in that class [Ref. 2:p. 404]. Inductive inference is the basis of inductive learning and is the process of generalizing assertions from specific observations about objects in classes of data. The inductive process takes an initial inductive hypothesis and develops assertions (rules) that can account for the observations of objects within the classes of objects.

There are two basic approaches to the inductive rule formulation, data-driven and model-driven. Model-driven methods expresses an a priori model of the data, in terms of production rules, and then tests these models against an actual data set. In data-driven methods, the data is first analyzed and then inductive models are developed to define the data set. The latter approach, data-driven models, is used for knowledge discovery in this research. This approach is seen as a more flexible means of developing rule structures, particularly since little a priori knowledge may be available about the data being analyzed. Additionally, true knowledge discovery of unanticipated results is more likely if the knowledge search is not hampered by constrictive biases introduced by a priori models.

Using a data-driven modeling approach, William Messier and James Hansen [Ref. 7:p. 1414] found considerable success in developing quality production rules for expert systems. They point out, however, there are a few limitations when using an inductive approach. First, the

data-driven method is difficult to apply to very large databases. Rule development is prohibitive when the availability and range of the variables are large and diverse. Second, inductive approaches can develop significant errors if the data set contains missing or erroneous values. Missing important instances or attributes may lead to rules that are misleading.

The ability to inductively generalize is an important process of knowledge discovery. The goal of inductive generalization methods in production systems is to find classes or a range of objects in the variable set that make the rules more applicable. More than one dependent variable category may be used to broaden the independent variables. The intent is to maximize the applicability of the attribute values. In rule induction, RHS dependent variables are broadened to include a wider range of the independent values. For example, a rule using a RHS dependent variable containing three categories, K_1, \dots, K_3 , could match a larger number of LHS independent variables (attributes) if the algorithm allowed for more than one RHS category to be included as an observation in the rule.

Generalization can also be accomplished by backing off from the exact values of the independent variables in order to gain a larger sampling of the data set, e.g., variable X may only apply to few instances in $(1 < X < 3)$, where expanding the range of X $(-100 < X < 500)$ may broaden the

applicability of X to a larger portion of the data set. Exact rules, on the other hand, limit the data sets to match one dependent variable and include only HIT-HIT situations.

The following section discusses a few of the early inductive rule production systems based on the data-driven approaches. As these systems developed they found techniques for dealing with the above problems and evolved into more powerful knowledge discovery systems.

1. CLS and ID3

An early inductive learning technique is the Concept Learning System (CLS) algorithm developed by Hunt et al (1966). The object of the algorithm is to take objects of a known class (categories of the dependent variable) described in terms of the attributes (independent variables) to generate a production system which classifies these given objects. In CLS a decision tree is developed by repeatedly segregating the data into smaller and smaller subsets. These subsets each held certain characteristics of the attributes that classified them into separate categories.

Quinlan (1979) eventually developed CLS into a more sophisticated program. This program, ID3, uses induction to develop a decision tree processes to classify and break the data down into a set of rules. ID3's inputs are a known class of data described in terms of a fixed set of attributes. ID3's output is a decision tree that classifies

the given cases of information. ID3 incorporates a top down induction approach using divide-and-conquer techniques to split the data into interesting attributes. The inductive algorithm constructs a rule by incrementally building a classification tree and repeatedly adding additional attributes to underspecified branches based on the estimated discriminatory power of that branch. This process continues iteratively until the data is fully partitioned.

ID3 is perhaps the most popular method of decision tree analysis because it is easy to implement and produces simple decision trees effectively [Ref. 8:p. 172]. However, as Parsaye and Hansson [Ref. 9:p. 141] point out, ID3 can easily generate too many decision trees with meaningless or uninteresting rules. Additionally, they find several other faults in ID3. First, ID3 is very sensitive to changes in the database. Small changes to the data can potentially produce large variations in the decision trees. Second, ID3 cannot generalize about the database. It produces exact decision trees, not general rules. Finally, similar to the second fault, ID3 cannot deal with inexact data and therefore cannot produce inexact recommendations.

2. AQ Algorithms

Parsaye's analysis of CLS and ID3 found that the decision tree approach to rule generation lacks robustness. Producing exact rules may help segregate the database into

unique subsets but severely cuts down on the ability to open up the data analysis to more useful general interpretations.

The AQ family of algorithms is another approach to knowledge discovery that sought to overcome the lack of generalization. These algorithms use both type and structural information of the database to generalize based on a given data samples. Constraints are applied to limit the focus search patterns. The basic idea is to select one example, generalize on the example to determine how much of the generalization applies to the database, ensuring the generalization doesn't violate the constraints. The Star system (Michalshi 1983) is a "knowledge generalization" system because of its ability to use an inductive and generalization process with applied user constraints. Parsaye discusses a definite problem with the AQ based algorithms programs when applied to large databases. Too many generalizations may be produced causing the value of the conclusions to be too "liberal." Additionally, the AQ algorithms could not deal with inexact data and tended to produce nonsensical rules.

The previous sections have discussed statistical inductive approaches to knowledge discovery. As pointed out, each approach has its own strengths and weaknesses. Statistical methods can be overly complex and difficult to use. On the other hand, inductive methods , such as CLS, ID3 and AQ may have a tendency to produce either too exact

or too liberal rules. A new approach is required that combines the strengths of both statistical and inductive methods while overcoming their inherent limitations. This new approach was introduced by Parsaye with IXL.

3. IXL/IDIS

Parsaye combined the statistical and inductive approach (machine learning) in his knowledge discovery program, IXL (Induction with eXtremely Large databases). IXL "...form(s)and test(s) various hypotheses about relationships in the database and uses machine learning algorithms to generate rules." [Ref. 10:p. 2] . At the heart of IXL, and the later version of the program IDIS, are two core modules. The statistical module analyses the data for statistical relationships and the artificial intelligence (AI) module interprets and tests these relationships then transposing them into easily understood rules [Ref. 10:p. 3]. The sophistication of IXL/IDIS comes into play when the program goes beyond sorting simple database descriptive statistics. By forming hypotheses on the statistical relationships, testing the hypotheses and repeating the process until patterns emerge, IXL/IDIS offers a clear alternative to the over generalization problems of CLS, ID3 and AQ algorithms. IXL/IDIS is able to circumvent the generalization problems that have plagued earlier knowledge discovery programs by allowing user defined

constraints on the AI module of pattern analysis. In addition, IXL has the ability to search large databases (in the hundreds of millions of records). For these reasons IDIS, the updated version of IXL, was chosen to benchmark the genetic programming approach developed at the Naval Postgraduate School.

IXL/IDIS has seen some success in commercial applications. The U.S. Army and Air Force exchange is an example where it has been applied to determine sales patterns based on the customer demographics. IXL helped them target sales and advertising towards the appropriate customer base. [Ref. 9:p. 157].

C. SUMMARY

This chapter we have reviewed the ability to capture large scale databases through mass storage and on-line databases connected by increasingly faster networks and the transfer of textual data to binary formats. Coupled with a vast increase in computer capacity, these innovations help deal with complex data and seek meaningful relationships among elements in the database. Recent attempts at knowledge discovery have shifted from traditional statistical techniques involving deductive analysis to more declarative techniques involving inductive analysis. Inductive algorithms, such as CLS and ID3, have shown great promise but are too restrictive to produce general rules. On the

other hand, AQ algorithms create too many generalizations producing many nonsensical rules. A new breed of systems (such as IDIS) that combine both statistical and inductive approaches are beginning to exploit the strengths of both approaches.

IV GENETIC ALGORITHMS / GENETIC PROGRAMMING

A. GENETIC ALGORITHMS

Genetic Algorithms (GA) emulate the natural science of evolution, the science that Darwin interpreted as the evolutionary process that adapted species to their environmental conditions. Genetic algorithms can be viewed as an adaptive search procedure which is based on the model of population genetics and natural selection. Koza defines Genetic Algorithms as follows:

Genetic algorithm is a highly parallel mathematical algorithm that transforms a set (population) of individual mathematical objects (typically fixed length character strings patterned after chromosome strings), each with an associated fitness value, into a new population (i.e. the next generation) using operations patterned after the Darwinian principle.
[Ref. 11: p. 18]

Genetic algorithms solve problems by evolving potential solutions through a process of randomly recombining the critical aspects of the problem. Problem conditions, actions or characteristics are typically represented as binary strings of data which are combined through genetic operations such as mutation or crossover with other possible conditions, actions or characteristics of the problem. Eventually, an optimal condition is achieved after numerous generations of "breeding" possible solutions. Some of the

basic terminology used in genetic algorithms follows [Ref. 11:pp. 18-26]:

1. Population: This represents the set of candidate solutions, or individuals.
2. Generation: Is one iteration of the genetic algorithm.
3. Crossover: Is the process where two individuals in the population exchange parts of their internal representation to create new individuals.
4. Mutation: Is the random change of part of the internal representation of an individual.
5. Fitness: Is the method used to select individuals from the populations through using quantitative measurement. The chosen individuals represent those that exist and survive in the population that may successfully reproduce.
6. Reproduction: Is the process by which part of the parent is taken, and put in the new generation without any alternation.

Genetic Algorithms operate on populations of data (individuals) where each represents a potential set of problem solutions. Individuals are selected during each generation by their "fitness" and then recombined through crossover. Offsprings may undergo mutation before they are inserted in the new population. This process continues until some specific termination criteria is met. Figure 1 shows a general flow chart of the genetic algorithm process operating on strings of individuals as depicted by John Koza [Ref. 11:p. 29].

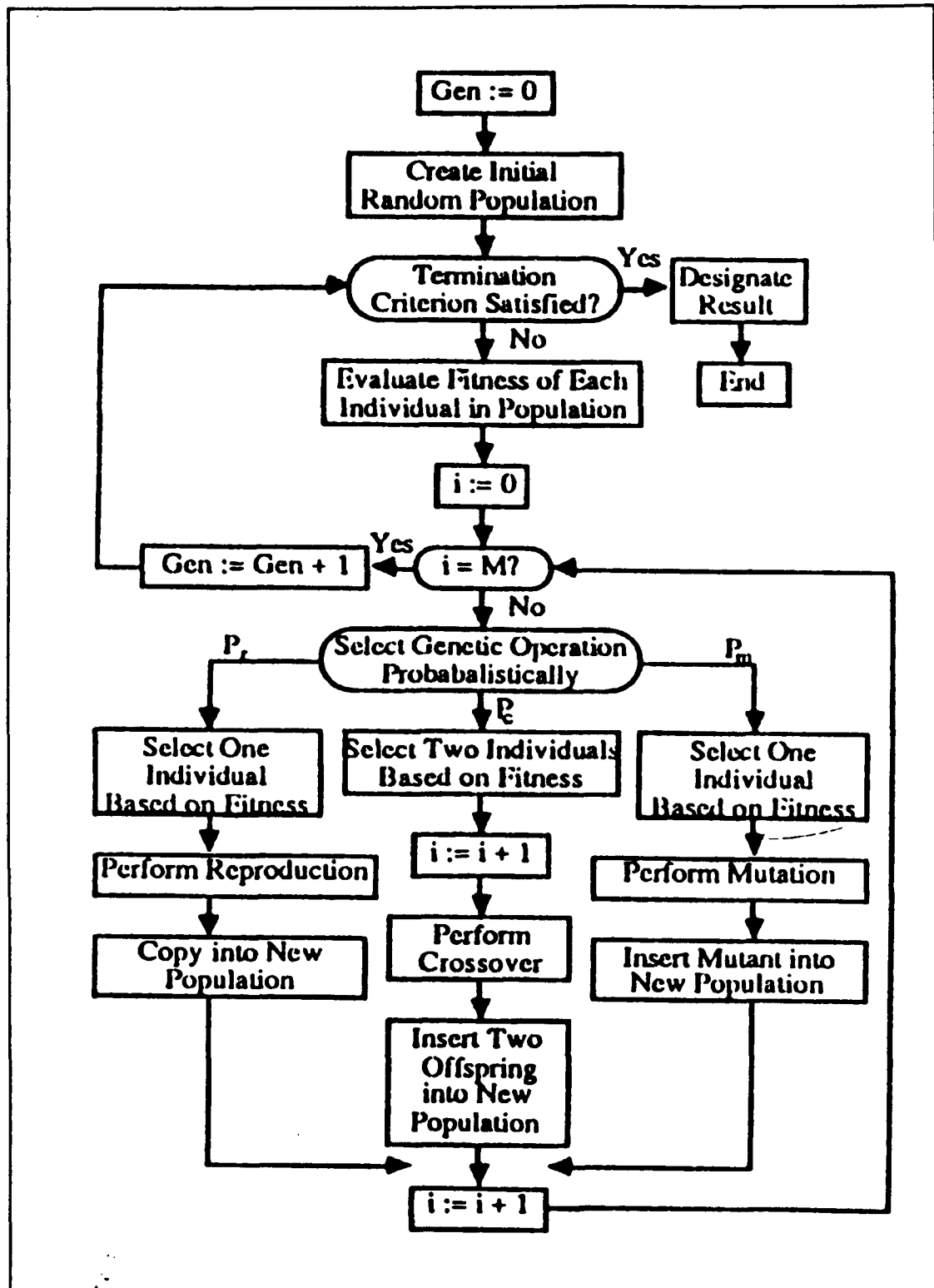


Figure 1 Flow Chart of the Genetic Algorithm Process

John Holland developed the genetic algorithm and provided its theoretical foundation in his book *Adaption in Natural and Artificial Systems* [Ref. 12:pp. 31-58]. Although the basics of Genetic Algorithms have been around since the mid 1960s [Ref. 13:p. 9], GAs have primarily been utilized in optimization and classification problems. Rule induction with genetic algorithms is a relatively new approach. Kenneth A. Dejong suggests using GA approaches to solve rule production systems. By "...maintain(ing) a population of candidate rule sets," GA can utilize the selection process to produce optimal rules [Ref. 14:p. 625].

B. SAMUEL: A RULE GENERATING GENETIC ALGORITHM

SAMUEL, standing for Strategy Acquisition Method Using Empirical Learning, is a genetic algorithm program developed at the Naval Research Laboratory. SAMUEL is designed to investigate behavior in simulation models. SAMUEL in essence is designed to learn rules for decision making agents. Coupling feedback mechanisms and performance evaluation techniques, SAMUEL seeks to learn optimal decision strategies through a set of rules that evolve over time. The goal of the program is to use a genetic algorithm to refine and improve upon an initial set of knowledge strategies, represented by a population of rules.

Knowledge is represented in SAMUEL through three levels of knowledge structures: *populations, plans and rules*.

Populations consists of a set of plans to deal with environmental simulations in the program. Plans, in turn, are composed of specific condition-action rules that are intended to represent performance strategies to deal with specific environmental conditions. Each rule consists of IF-THEN conditional statements representing a set of conditions and appropriate responses to those conditions (actions). An example rule might be:

Rule 10

IF range = [250, 1000] and speed = [500, 1200]

THEN SET turn = [0, 90]

Performance evaluation of the competing plans is accomplished through the GA. Each generation contains a population of plans that compete and are measured by a *fitness* function. Based on the relative grades assigned by the fitness evaluation, the GA selects the best of the high performing plans for additional reproduction, crossover and mutation. The reproduction and testing cycles repeat until user specified criteria are meet.

The advantage of SAMUEL is that it provides an important framework for developing a GP that analyses data to generate and evaluate production rules. SAMUEL is limited by the restrictive rule structure of IF-THEN statements. SAMUEL is also limited to rule structures of fixed length and does not incorporate combinations of logical operators, such as AND,

OR and NOT. These limitations on rule structure are removed by genetic programming.

C. GENETIC PROGRAMMING

For many problems in machine learning and artificial intelligence, the most natural known representation for a solution is a hierarchical computer program of indeterminate size and shape. [Ref. 11:p. 210]

Genetic programming is a variant of genetic algorithms with a different problem representation. The basic approach to genetic programming is summarized by John Koza in three steps as follows [Ref. 11:p. 213].

1. Generate an initial population of random compositions on the functions and terminals of the problem (computer programs).
2. Iteratively perform the following substeps until the termination criterion has been satisfied:
 - a. Execute each program in the population and assign it a fitness value according to how well it solves the problem.
 - b. Create a new population of computer programs by applying the following two primary operations.
 - (i) Copy existing computer programs to the new population.
 - (ii) Create new computer programs by genetically recombining randomly chosen parts of the two existing programs.
 - (iii) These operations are applied to computer program(s) in the population chosen with a probability based on Darwinian fitness.
3. The single best computer solution in the population is designated as the result of the genetic program. The result may be a solution (or an approximate solution) to the problem.

The application of genetic programming is relatively new. For the most part these applications have focused on optimization and classification problems unrelated to generating production rules. [Ref. 15:p. 12].

V NPS GENETIC PROGRAM (NPSGP)

A. BACKGROUND

The Naval Postgraduate School Genetic Program (NPSGP) adapts Walter Tackett's [1993] implementation of John Koza's genetic program developed at Stanford University. NPSGP supports rule generation. Constraints can be added to the structures evolved by the GP. Constraints are used to define the formats for the rules to be generated by the system. A format specified in NPSGP dictates that rules follow the (if X then Y) patterns. In these rules, the dependent variable is on the right hand side (RHS) and independent variables are on the left hand side (LHS). In genetic programming these independent variables are called terminal sets. Before going further into the details of the NPSGP system, the data set used to evaluate this program is described. This will assist in explaining the functions of NPSGP.

1. Data Description

The data set used to demonstrate NPSGP contains quantitative data, specifically, the basic components of U.S. Gross Domestic Product (GDP) for the years 1929 to 1991 [Ref. 16:p. 341-474]. The components include Personal Savings, Disposal Personal Income, and Net Exports/Imports.

Other relevant variables were also used, such as civilian unemployment rates and 10-year Government Bond rates. Finally, some of this data was converted into ratio's , e.g., Personal Consumption Expenditures were divided by GDP.

GDP Growth, the dependent variable, was defined as a change in GDP and was categorized into four discrete classes. The first category was assigned a value of high positive (HPOS) if Indexed GDP grew by more than 2.5 per cent from the previous year. Indexed GDP growth from 0 to 2.5 per cent was categorized as low positive (LPOS). GDP growth of 0 to -2.5 per cent was categorized as low negative (LNEG). Any negative growth greater than -2.5 per cent was classified as high negative (HNEG).

The function set consisted of the operators in the rule structures (i.e., IF, OR, AND, NOT). The rule is represented by a collection of "terms" where each term specifies attribute-value ranges ,called selectors. As an example, a selector including the attribute UNEMPLOYMENT may have an applicable range of 0 to 25 per cent in the database; $0 < \text{UNEMPLOYMENT} < 25\%$.

Such a strategy offers robust rules through compensatory selection; the terminal set selected by one variable can compensate for by the terminal set selected by another variable. For example, the following rule offers a compensatory strategy for either the change in 10 Bond rates

or the change in personal savings and gross public debt as a percentage of GDP:

```
GDP GROWTH = LPOS
IF
    2.2 < CHANGE IN THE 10 YEAR BOND RATE < 9.20
OR
    0.95 < GROSS PUBLIC DEBT AS A % OF GDP < 2.16
AND
    1.35 < CHANGE IN PERSONAL SAVINGS < 2.61
```

B. CONTROL PARAMETERS

NPSGP offers a number of control parameters that the user can adjust to tailor the program to the type of data and analysis.

1. Fitness Measure

The fitness measure is the quantitative evaluation of how well the rule matches the data. NPSGP uses a contingency table approach to establish a method of determine if the LHS of the rule classifies a RHS category, e.g., HPOS, LPOS, LNEG, or HNEG. First, an observation of the independent variable either belongs to a category (C+) or doesn't belong to that particular category (C-). Second, the LHS of the rule may be true for an observation (M+) or false (M-). The following contingency table shows the four possible conditions:

		RIGHT HAND SIDE	
		C+	C-
LEFT HAND SIDE	M+	A	B
	M-	C	D

A, B, C, D represents the total number of data points in each subset. For example, "A" represents the number of cases where the LHS and the RHS are true for an observation; this is a HIT-HIT condition. "B" represents number of cases where the LHS is true but the RHS is false; this is the MISS-HIT. "C" represents a HIT-MISS condition where the RHS is true but the LHS is false and "D" is the MISS-MISS condition where both the LHS and RHS are false.

A fitness function may be defined in terms of the various cases in the contingency table. For example, the simplest method for computing fitness is to divide the MISS-HIT value B by the HIT-HIT value A. To eliminate the possibility of division by or into zero, a small number is added to each of the parameters, i.e., $(B+1)/(A+1)$. NPSGP minimizes the value of the fitness function.

The following two examples illustrate how fitness function are used to evaluate rules. In both cases there are 63 data points for each attribute. The first case assumes that a rule produces 38 instances of HIT-HIT, category A, no instances of category B, MISS-HIT, and no instances of category C, HIT-MISS. By adding the constant

value of 1 to each of the parameters the following fitness measure is calculated.

$$\begin{aligned}\text{Fitness} &= (B + C + 1) / (A + C + 1) \\ &= (0+0+1) / (38+0+1) = 0.025\end{aligned}$$

In the second example, a rule produces 30 instances of HIT-HIT, category A, and 8 instances of MISS-HIT, category B, and 3 instances of MISS-HIT, category C. The fitness calculation then will yield:

$$\begin{aligned}\text{Fitness} &= (B + C + 1) / (A + C + 1) \\ &= (8+3+1) / (30+3+1) = 0.3529\end{aligned}$$

The second rule did not fit the data set as well as the first rule. It had 8 cases where LHS rule fit the data set (interest rates, savings rates, etc.) but not the RHS dependent variable (GDP growth = HPOS; LPOS; LNEG; HNEG) and 3 cases where the RHS rule fit the dependent variables but not the data set. The higher fitness measure indicates the fit was not as good.

2. Selection Criteria

Tournament selection is a method used to select the best fitting rules from the total generated population. It is similar to the way a winner is chosen from a tournament

among competitive teams. In NPSGP tournament selection was set at a value of six, resulting in a grouping of six rules from which the best is then chosen. Another method is a probabilistic selection method where the probability of selecting an individual depends on its fitness. For example, the best individual in the population may have a high probability (close to 1.0), while individuals in the mid-range may have a probability of approximately 0.5. The worst individual of the group will have a probability of 0. [Ref. 11:pp. 604-607].

3. Population and Generation

NPSGP also allows the user to select the population size to be specified. A population of 1,000 in this case means 1,000 rules will be initially randomly generated and then competitively adapted through crossover. Users can also select the total number of generations to be run, which serves as the termination point of the program. The total number of generations and population size required to produce an optimal solution may be determined by trying several combinations of these parameters.

C. EVOLUTION OF RULES

The program starts with an initial population of a number of randomly generated rules composed from the function and terminal variables sets. This initial population will usually contain a high percentage of poorly

fitting rules that are later refined by the crossover process. Reproduction and crossover operations are then applied to breed a new population of offspring rules. This process continues until the termination criteria is satisfied. Figure 2 outlines the major processes that the program follows in generating rules.

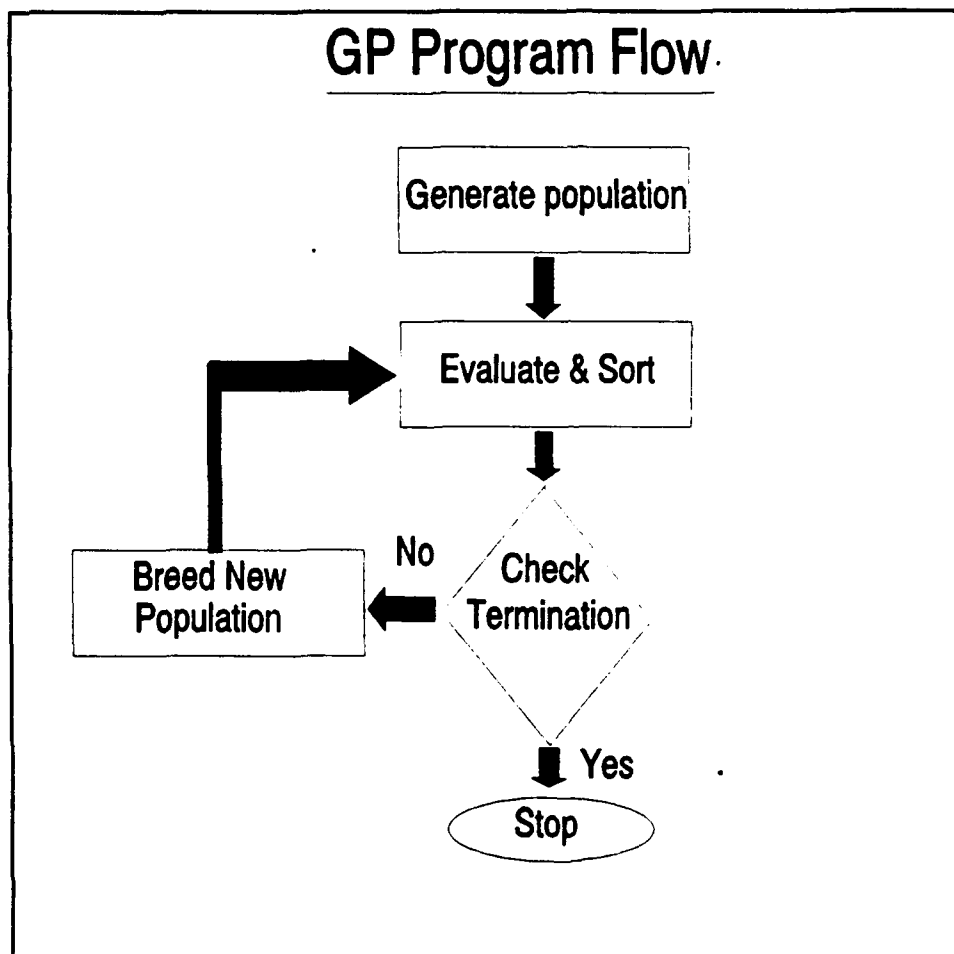


Figure 2 - GP Program Flow

D. AN EXAMPLE OF THE GP PROCESS

The following illustrates the NPS genetic program output using the econometric data outlined in Appendix A. Program control parameters were set with a population size of 3000, crossover rate of 75 per cent, and termination parameter of 50 generations. The fitness measure $(B + C + 1)/(A + C + 1)$ provides an initial means to minimize the number of HIT-MISSES and MISS-HITS. Although programmed in a "C" language shell, NPSGP produces rules using the LISP list format. Using the above fitness measure, NPSGP reported the best rule, e.g., the rule with the lowest fitness value, in generation 0 as follows:

```
Best of Generation 0:
(IF
  (AND
    (TWIXT
      4.760321
      UNEMPCIV
      -3.134517)
    (TWIXT
      1.770329
      BOND10YR
      -15.607920))
  3.000000)
Number of records matched by LHS: 2.000000
Number of misclassified records: 0.000000
Confidence: 1.000000
Validation Fitness= 0.333333
```


The rule can be easily translated to a more understandable format. The above rule translates to:

```
IF
  -3.134517 < UNEMPCIV < 4.760321
AND
  -15.607920 < BOND10YR < 1.770329
THEN
  GDP GROWTH = HNEG
```

As indicated above the LHS of the rule matched two observations in the data set. The confidence factor, used as a measure of exactness of a rule, is calculated by the formula $A / (A + B)$. This formula calculates the percentage of correct RHS classifications to total RHS classifications, when all LHS instances match the data set. In this example there were no miss-classifications the formula computed to $2 / (2 + 0)$ or 1.0, representing 100 per cent confidence. The rule was exact in that it correctly classified all the records pertaining to that rule into a HIT-HIT situation. The fitness was computed as $[B(0) + C(0) + 1] / [A(2) + C(0) + 1] = 1/3 = 0.33333$. Constant values of 1 were added to the fitness denominator and numerator to avoid division by zero. Finally, the numeric code 3 at the end of the rule represents the RHS category, e.g., $GROWTH = HNEG$.

Better rules were evolved to fit the data set through generations of reproduction. As an example, the best rule of generation 3 follows:

Best Rule of Generation 3:

```
(IF
  (AND
    (TWIXT
      0.341340
      GDBTPERGDP
      -2.211879)
    (TWIXT
      -4.860248
      GDPIPERCHG
      0.170821))
  0.000000)
Number of records matched by LHS: 32.000000
Number of misclassified records: 3.000000
Confidence: 0.906250
Validation Fitness= 0.315789
```

Translated:

```
IF
  -2.211879 < GDBTPERGDP < 0.341340
AND
  -4.860248 < GDPIPERCHG < 0.170821
THEN
  GROWTH = HPOS
```

The improvement in fitness, the NPSGP method of gauging rule improvement, is noted by the decrease from 0.333333 to 0.315789. This rule applies to different LHS attributes and RHS category. Finally, the rule has 32 HIT-HIT classifications and 3 MISS-HIT. Here the fitness formula used, $A/(A+B)$, moved the GP toward more generalized rules.

By generation 7 the fitness measure "optimized", the validation fitness value of the best rule minimized at .263158. As shown in the example below, this rule has a net decrease of one MISS-HIT observation from the above example.

Best Rule of Generation 7:

```
(IF
  (AND
    (TWIXT
      -6.015694
      PFDBTGDP
      0.895278)
    (TWIXT
      0.239258
      GDPIPERCHG
      -24.963970))
  0.000000)
Number of records matched by LHS: 32.000000
Number of misclassified records: 2.000000
Confidence: 0.937500
Validation Fitness= 0.263158
```

Translated:

```
IF
  -6.015694 < PFDBTGDP < 0.895278
AND
  -24.963970 < GDPIPERCHG < 0.239258
THEN
  GROWTH = HPOS
```

Through crossover NPSGP was searching for the best rule that could fit both the RHS categories and LHS terminal set. Although the best of breed fitness measure did not improve after this generation, changes continued in rule attributes of subsequent generations. For example, the top rule in generation 50 included the attributes GDBTPERGDP and GDPIPERCHG. It had the same number of classifications/misclassifications and confidence as the best rule in generation 7. However, the rule in generation 7 contained the attributes PFDBTGDP and GDPIPERCHG.

This improvement of the fitness measure is graphically illustrated in Figure 3. With the relatively small data set used in this application, the fitness measure very quickly converged to an optimal value.

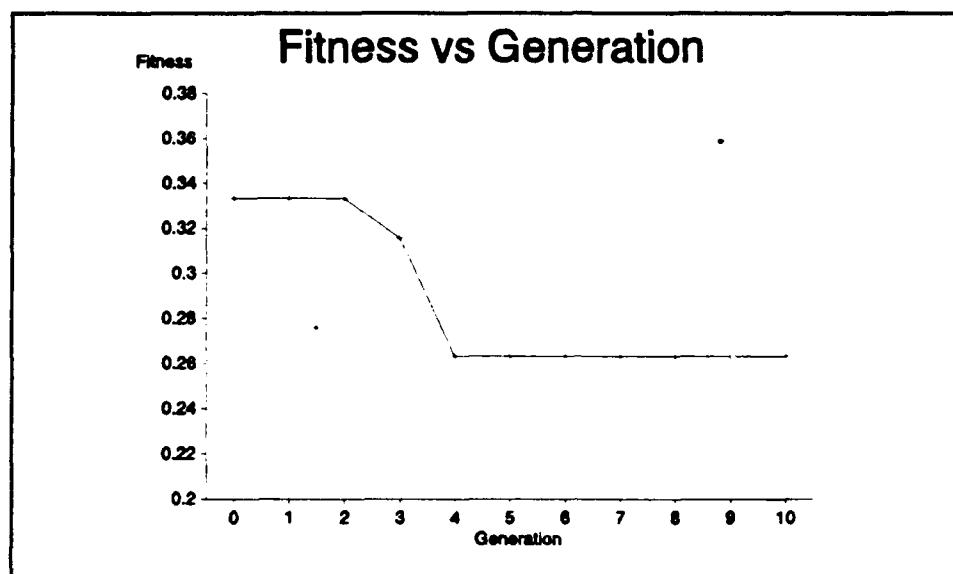


Figure 3. Fitness verses Generation

E. GP EVALUATION RUNS

To test how sensitive the fitness measure is to changes in parametric inputs, the GP was run with different population sizes and crossover rates using the same fitness measure $(B + C) / (A + C)$. First, the crossover rate was varied holding the population constant at 3,000. Crossover rates of 60, 65, 70, 75, 80, 85, 90, 95 and 100 per cent were tried. The results indicate that changes in the

crossover rate had a minimal effect on fitness. In all cases, the fitness of the "best of generation" optimized in generation 7 with a value of 0.263158. This result may be due to the relatively small size of the database. A larger data set using real and discrete terminal values may have produced quite different results.

Another test was performed changing the population size to determine its affects on fitness convergence.

POPULATION	500	1,000	2,000	3,000	5,000	10,000
OPTIMAL GENERATION	12	8	15	7	11	16
OPTIMAL FITNESS	0.263158	0.263158	0.236842	0.263158	0.184211	0.038462

Table 1. Fitness vs Population

As Table 1 indicates, the size of the population did affect the optimal fitness and the number of generations required to reach it. Increasing the population from 3000, to 5000, and then 10,000, lowered the fitness value. This indicates there is a minimum population size to produce the best rule results. The larger the selection pool the better statistical odds of producing rules that best model the data. The fitness measure (0.08333) of generation zero

(without crossover) using a population of 10,000 was better than the optimal fit of the smaller population after generation 7.

F. SUMMARY

Applying NPSGP to the data set revealed some important characteristics of GP rule generation. First, the composition of observations used in the fitness function is critical to the type and depth of rules produced by the GP. For example, the fitness function can be modeled to either produce generalized rules that include both HIT-HIT and MISS-HIT observations, or, it can produce exact rules using only HIT-HIT observations. This point is elaborated further in the next chapter.

Second, it is important to review all rules produced by the GP that meet a desired minimum fitness measure due to the differences in the attributes and data range variations of those rules. Interesting rules and valuable data associations can be overlooked if users only focus on those rules where the program has minimized the fitness value. Third, population size has a significant affect on rule generation, including both fitness values and LHS attribute ranges.

VI COMPARISON OF KNOWLEDGE DISCOVERY RESULTS

This chapter uses the data outlined in Appendix B to compare two known knowledge discovery approaches, i.e., statistical and inductive, against the NPSGP. The comparison is based on producing data element associations between the RHS categories and LHS attributes. As mentioned in Chapter III, IDIS was selected as representative of the inductive knowledge discovery approach because of its ability to overcome the limitations of other inductive systems. While there are numerous statistical methods for discovering data element associations, stepwise discriminate analysis (SDA) was used to select which LHS attributes most significantly discriminated between the RHS categories. The SDA was done using the SAS statistical package and is described in greater detail in the following sections.

The benchmark for comparing all three approaches, was the ability to identify associations among LHS data elements. SDA accomplishes this by identifying statistically significant independent variables that discriminate between dependent variable categories. This SDA list of significant variables is then compared to the LHS attributes produced in the NPSGP rules. The rationale behind this comparison is to determine if NPSGP could

develop data element associations that include variables found to be significant by SDA.

Benchmarking IDIS to NPSGP meant reviewing the generated rules to compare LHS attributes. Any attribute listed in a rule was assumed to be significant. The LHS attribute ranges were not reviewed in detail nor was there an attempt to determine the usefulness of the rules.

A. STATISTICAL ASSOCIATIONS

SAS discriminate analysis procedures analyze data sets containing one dependent variable and several independent variables. The purpose of this discriminate analysis is to find the subset of those quantitative independent variables that are statistically "important" and can reveal differences among the dependent variable categories. However, SDA is not necessarily the best approach for all purposes. It has to be used carefully and reflect the user's knowledge of the data. [Ref. 6:p. 911].

Within the SAS discriminate analysis, the STEPDISC function was chosen to produce a discrimination model by selecting a subset of the quantitative variables based on one of two following criteria:

1. Analysis of covariance using the F-test to determine significance. The variables chosen act as covariants to the dependent variable (GDP Growth).

2. Analysis of the squared partial (R^2) correlation value between the dependent variable (GDP Growth) and the independent variables, controlling for the effects of other variables selected for the model. [Ref. 6:p. 909].

Stepwise selection begins the process without any variables in the model. Variables are included into (or later excluded from) the model if they contribute to the discriminatory power of the model as measured by Wilke's Lambda. This process is repeated in steps until all variables meet either the criteria to stay or be dropped from the model. There is one potential weakness of this model: relationships between the unselected variables are not analyzed. This potentially excludes statistically significant variables. [Ref. 6:p. 910].

1. Results:

The results of using SDA on the 16 independent variables listed in Appendix B is illustrated in the Table 2. Table 2 first segregates the significant and non-significant independent variables in the model then lists these variables by decreasing order of partial R^2 value. The R^2 values represent the one-way analysis of covariance between the selected variable and the remaining variables not chosen for the model.

VARIABLE SELECTED	VARIABLE NOT SELECTED	PARTIAL R ²	F STATISTIC	PROB > F STATISTIC
UNEMPCHG		0.6217	32.316	0.0001
INDXGPDIPER		0.2276	5.697	0.0017
EXPPERGDP		0.2193	5.5151	0.0033
PCEPERCHG		0.2079	4.986	0.0039
POPCHGPER		0.1379	2.880	0.0443
GDPPIPERCHG		0.1092	2.125	0.1082
BOND10YR		0.1066	2.228	0.0950
SAVPERDPI		0.1054	2.0817	0.1138
	UNEMPCIV	0.0935	1.753	0.1679
	GDBTPERGDP	0.0837	1.553	0.2122
	IDXPCEPER	0.0377	0.665	0.5773
	DEFPEROUT	0.0312	0.548	0.6517
	PFDBTGDP	0.0252	0.439	0.7263
	SAVPERCHG	0.0190	0.329	0.8046
	CHGBDRATE	0.0172	0.298	0.8264
	GDEBTGDP	0.0070	0.120	0.9480

TABLE 2. Discriminate Analysis Results of Data Set

The discriminate analysis shown in Table 2 indicates that eight of the sixteen attributes were significant and can be used to discriminate between the four categories of the dependent variable (GDP Growth = HPOS, LPOS, LNEG, HNEG).

Error rate estimation (probabilities of misclassification) measures the discriminate model's performance. As shown in Table 2, five of the eight variables included in the model had error estimates (Probability > F Statistic) of less than 5 per cent while six of the eight nonselected variables had error estimates of greater than 50 per cent. Overall the analysis shows a strong discriminative association in eight of the sixteen independent variables. These results will be used to test the robustness of the rules produced using NPSGP and the same LHS attributes.

B. IDIS

As mentioned in Chapter III, IDIS uses a search algorithm blending statistical analysis with inductive heuristic learning. This algorithm is fixed within the program and search flexibility comes in setting the search control parameters, shown in Appendix B. These parameters allow IDIS to focus on generating rules from either narrow or broad search patterns. For example, liberalizing the program control parameters (confidence factor, error margin,

data range generality and minimum required records per rule) will produce rules with multiple LHS attributes and broad data value ranges. These rules may be too broad to be "interesting" and require further refinement. To focus these rules, control parameters are narrowed to reduce the selected LHS attribute range values, that is reduced attribute and range generalization. On the other hand, IDIS can also search for specific dependent variable categories. However, this may come at the expense of rule generalization and error estimation, particularly if the category represents only a small portion of the data set.

Initially, a broad approach was used by selecting control parameters so that IDIS would search for a wide variety of rules linking LHS attributes to RHS categories. For example, control parameters were set at a certainty factor of 80 per cent, 25 per cent error margin, a minimum of 6 records per rule, 100 per cent range generalization and the dependent variable was set to produce rules for any of the RHS dependent categories (HPOS, LPOS, LNEG, HNEG). However, these control parameters were too broad and only produced rules for HPOS. To search for rules in the other RHS categories, the control parameters were subsequently relaxed to include additional data elements. In addition, the dependent variable categories were focused on LPOS, LNEG and HNEG. The results of these searches are discussed below.

1. IDIS Results

Initially IDIS produced rules including every LHS attribute, either singularly or in combination with other LHS attributes, but for only one dependent variable category, GDP GROWTH = HPOS. IDIS produced rules for GDP GROWTH = LPOS and LNEG only after restricting the RHS search to only those categories. The inductive rules for these categories are not as "good" as those for HPOS. Further, it was necessary to reduce the minimum record constraint to 3 and increase the margin of error to 50 per cent before the system produced any rules at all. The only LPOS rule contained three attributes; this indicates that IDIS could not find a singular or dual attribute rule for LPOS. This singular rule is shown below:

```
GDP GROWTH = LPOS
IF
  0.3      <= GDPIPERCHG <= 7.6
AND
  0.011    <= POPCHGPER  <= 0.0208
AND
  -0.1843 <= GDBTPERGDP <= -0.0039

CONFIDENCE FACTOR = 66.67 %
MARGIN OF ERROR   = 36.4 %
NUMBER OF APPLICABLE RECORDS = 9
```

The only rule produced where GDP GROWTH was LNEG follows:

```
GDP GROWTH = LNEG
IF
  1.1    <= UNEMPCHG <= 2.9

CONFIDENCE FACTOR = 63.64 %
MARGIN OF ERROR = 33.0 %
NUMBER OF APPLICABLE RECORDS = 11
```

These results imply that a fixed search algorithm may not identify all potential rules in the RHS categories. The IDIS user may need to force the search for different RHS categories. As a tradeoff to finding these rules the user may be required to accept lower confidence factors and higher error estimates.

C. NPSGP

NPSGP uses a random and evolutionary approach (through adaptive competition). There is no fixed search pattern built into the initial rule generation process. Unlike traditional methods which sometimes presume a specific structure (i.e., linear compensatory or correlated attributes), NPSGP starts off with a set of randomly selected rules [Ref 3:p. 219]. Structural bias may be introduced as part of the fitness function that discriminates between these rules in later generations. NPSGP offers flexibility to focus the data mining because it

is easy to change the fitness function. The fitness function's effect on rule selection and along the NPSGP program parameters used in this research are discussed further in the following sections.

1. NPSGP Parameters

For examples discussed in this chapter, NPSGP control parameters were set to the following values:

1. Population size: 5,000.
2. Crossover rate: 75 per cent.
3. Random generator seed value: 4.
4. Function Set was originally limited to one conjunction (AND) and later modified to include an arbitrary number of logical operators (AND, OR and NOT).

Population was set at 5,000 to limit CPU time and memory used in the analysis. As indicated in chapter V, crossover had a small effect on GP rule generation. An arbitrary crossover value of 75 per cent was assigned. Introducing of a random seed starts the random number generator used in selecting the initial rules. Changing the random generator seed number has a small effect on the types and ranges of the rules generated. However, these differences are expected to be normalized over a number of runs. In addition to the above parameters, NPSGP also allows the user to select the number of rules N , it shows at the end of each generation. These "top N " rules are listed in descending order of computed fitness value. To limit the

rule review process we set the list to show only the top 30 rules for each generation.

2. Fitness Functions

This section illustrates how changing the fitness function can refocus rule discovery. Ultimately, this leads to different knowledge associations (rules) than either the statistical discriminate approach or the combined efforts of the singular search function deductive/inductive IDIS approach.

A variety of fitness functions were tested in NPSGP, most yielding different sets of rules. To illustrate the impact that changing the fitness function has on the rules generated, the following two fitness functions were selected:

$$Fitness = 0.8 - \frac{A+1}{A+B+1} \quad (1)$$

$$Fitness = \frac{0.25B + 0.5C + 1}{2A + 0.5B + C + 1} \quad (2)$$

The intent of fitness function (1) was to force the system to generate rules that are correct approximately 80 per cent of the time. Since NPSGP minimizes the fitness

value, subtracting the calculated value of $(A / (A+B))$ from 0.8 (80 per cent) will drive the value of the confidence equation $(A / (A+B))$, to 80 per cent. Therefore, the fitness function (1) will be optimized when $(A / (A+B))$ approaches 0.8 (assuming it is ≤ 0.8). When $(A / (A+B))$ is greater than 0.8, the fitness is set at an arbitrary high number so that the system will avoid such solutions.

Fitness function (2) uses a weighted approach that assigns relative importance to the different observations, e.g., A, B, C, D. This approach demonstrates that rule search can be focused for a specific mix of observations. This increases rule exactness or generality, as desired by the user. For example, when the aim is to produce exact rules, observation A (HIT-HIT) receives a greater weight than the other observations. To broaden the rule applicability to the RHS, observation B (MISS-HIT) is assigned greater numeric weight. To produce LHS rule generality, observation C (HIT-MISS) is given more weight.

Fitness function (2) uses this weighted approach to introduce more LHS and RHS generality into the rules. This is accomplished by emphasizing A (HIT-HIT), limiting emphasis on C (HIT-MISS), de-emphasizing B (MISS-HIT) and ignoring D (MISS-MISS). The following sections apply these two fitness functions to the data set.

3. Fitness Function Results

To produce simple rules, both fitness functions were used with function set limited to only one conjunction (AND) with no more than two attributes. These rules are shown in examples A and B below. Later, the function set was relaxed to an unlimited number of the conjunction "AND" leading to rules shown in example C.

a. Example A:

Example A contains four samples of the rules produced by NPSGP, one for each of the four categories of the dependent variable GDP GROWTH. Fitness function (1) was used as the discrimination factor.

1. Generation 42, Rule 8:

GDP GROWTH = HPOS

IF

-6.81 < GDPIPERCHG < -2.53

AND

-2.53 < POPCHGPER < 3.35

FITNESS = 0.005128

CONFIDENCE = 78.95 %

APPLICABLE RECORDS = 38

2. Generation 0, Rule 8:

GDP GROWTH = LPOS

IF

1.47 < CHGBDRATE < 7.80

AND

-1.45 < GDPIPERCHG < 1.58

FITNESS = 0.05000

CONFIDENCE = 66.67 %

APPLICABLE RECORDS = 3

3. Generation 0, Rule 21:
 GDP GROWTH = LNEG
 IF
 -1.79 < CHGBDRATE < 0.39
 AND
 -0.79 < GDPIPERCHG < 2.33
 FITNESS = 0.05000
 CONFIDENCE = 66.67 %
 APPLICABLE RECORDS = 3
4. Generation 0, Rule 9:
 GDP GROWTH = HNEG
 IF
 -3.73 < PFDBTGDP < 1.76
 AND
 -3.19 < SAVPERDPI < 1.24
 FITNESS = 0.05000
 CONFIDENCE = 66.67 %
 APPLICABLE RECORDS = 3

As these examples illustrate, the top 30 rules produced using fitness function (1) indicated rules involving all four RHS categories.

b. Example B:

Example B is a sample rule for GDP GROWTH = HPOS using the weighted approach (i.e., fitness function (2)). Unlike fitness function (1), the RHS categories (LPOS, LNEG, HNEG) in the top 30 rules.

1. Generation 12, Rule 1:
 GDP GROWTH = HPOS
 IF
 0.34 < GDEBTGDP < 3.53
 AND
 -1.10 < PCEPERCHG < 21.08
 FITNESS = 0.034161
 CONFIDENCE = 84.44 %
 APPLICABLE RECORDS = 45

c. Example C:

The following rule is representative of the rules produced with fitness function (2) without the singular conjunction restraint. Again, rules including other categories (HPOS, LNEG, HNEG) were not among the top 30 rules.

```
1. Generation 30, Rule 1:
   GDP GROWTH = LPOS
   IF
       3.99 < UNEMPCIV < 5.84
   AND
       -1.55 < SAVPERCHG < 0.77
   AND
       -2.10 < PCEPERCHG < 2.36
   FITNESS = 0.106061
   CONFIDENCE = 100.00 %
   APPLICABLE RECORDS = 7
```

4. Function Set Results

In the above examples, rule search patterns have focused on simple rule associations using one conjunction (AND), both for IDIS and NPSGP. Knowledge about complex data associations is not adequately represented using only singular (AND) conjunctions. Rules representing complex associations between data elements can be improved by widening the choice of logical operators to include (OR and NOT). For example, the logical operator "OR", provides rules where there is choice between equivalent attributes (IF X or Y THEN Z). The operator "NOT" is used to depict exceptions in the rule (IF X and NOT Z, THEN Y).

To illustrate, the program using function set (1) was modified to permit unlimited combinations of logical operators (AND, OR, NOT). NPSGP was able to produce rules using all three logical operators. Example D, demonstrates this additional flexibility. It provides a sample of the rules NPSGP produced using multiple logical operators. IDIS dose not provide this ability.

a. Example D:

```
1. Generation 4, Rule 1:
  GDP GROWTH = LPOS
  IF
    1.84 < CHGBDRATE < 5.69
  AND
    -1.56 < GDPIPERCHG < 2.23
  OR
    3.98 < UNEMPCIV < 5.41
    OR NOT
    0.49 < UNEMPCIV < 15.99
  AND
    -0.40 < GDPIPERCHG < 4.116
  FITNESS = 0.02222
  CONFIDENCE = 75.00 %
  APPLICABLE RECORDS = 8
```

D. COMPARISON OF RESULTS

Comparing the data element associations developed by the three different approaches brought out two significant differences. First, the ability to focus the search algorithms (i.e., "fitness function"), to force the search pattern toward exact or general patterns is the most important component of any knowledge discovery program. If the algorithm can not be focused, the knowledge discovery

program is locked into the fixed search pattern defined by that algorithm.

With IDIS, control parameters used in conjunction with the search algorithm can be changed. However, this does not change the fixed search algorithm itself. This limitation made it difficult for IDIS to find rules for the RHS categories LPOS, LNEG and HNEG. NPSGP was able to discover some rules in these categories using fitness function (1).

Additionally, the discriminate analysis function was limited to statistical association tests, such as the F test, R^2 partial correlation and Wilke's Lambda tests, to find data element associations between dependent and independent variables. This stepwise discriminate data analysis eliminated some interesting attributes from further analysis. These attributes were used to produce rules in both IDIS and NPSGP. For example, NPSGP developed rules using the eight significant variables in the discriminate analysis. It also developed rules with a few of the variables classified as non-significant (e.g., Example C included SAVPERGHG, Example B included GDEBTGDP and Example D included CHGBDRATE).

For the second difference, including additional logical operators (NOT and OR) increases robustness in rule generation. It provided both exceptions to the rule and/or equivalent attributes within the same rule. Due to it's limited rule structure, IDIS does not include additional

logical operators. It is limited to the single conjunctive "AND." NPSGP's flexibility to include "OR" and "NOT" operations leads to more complex rules and associations that allow deeper data mining with complex data element associations. .

VII CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The focus of this research is to compare several methods of knowledge discovery according to their ability to identify associations in data elements. One of the methods chosen for this purpose was a statistical approach, discriminate stepwise analysis. Stepwise discriminate analysis identifies independent variables that statistically discriminate against various dependent variables. The second method IDIS, combines statistical and inductive methods. IDIS represents associations between data elements by production rules. The last method is a genetic program that evolves rules through competitive adaption.

This analysis compared the structural characteristics of the rules generated, rather than their usefulness. It should be noted that the definition of usefulness is very subjective and depends heavily on the purpose for which the rules were generated. The test data set used in this study combined Gross Domestic Product information and economic indicators.

Several findings resulted from the analysis. First, the NPS GP model was able to develop rules involving more RHS categories than IDIS. In one GP program run (Example A in

Chapter VI), rules were produced for all four dependent variable categories. IDIS produced an abundance of rules using all attributes for HPOS. However, IDIS was unable to produce any rules for HNEG. Relaxing IDIS's constraints (control parameters) to very low confidence levels and high error estimates only produced one rule LPOS. NPSGP also found rules with more LHS attributes. Further more, the stepwise discriminate analysis found eight statistically significant LHS attributes. NPSGP found rules involving three additional LHS attributes; SAVPERCHG, GDEBTGDP and CHGBDRATE.

Second, NPSGP demonstrated that crossover had a limited effect on the GP's rule generating capabilities for small quantitative data sets (16 attributes by 63 instances). Population size was more important in GP generated rules, as measured by the fitness function. Population size primarily affected the selection and range of the attributes chosen.

Third, fitness function (search algorithm) is the most important parameter used in knowledge discovery programs. It sets the search focus for selecting independent and dependent variables. The fitness function is used as a filter to select rules that fit into a particular schema, e.g., exact or general rules. IDIS and stepwise discriminate analysis are locked into a fixed search pattern. The flexibility to change the fitness function

(search algorithm) gives the NPSGP program a significant advantage.

Finally, the ability to expand the function set by adding of logical operators such as "OR" and "NOT" gives NPSGP the potential to handle complex rule associations not covered by IDIS or statistical methods.

B. FUTURE RESEARCH

Recent developments in genetic programming offer great potential for knowledge discovery. GP is not restricted to a priori deductive and inductive paradigms that limit the scope of knowledge discovery. It offers a radical breakthrough that cuts through limitations of traditional approaches. NPSGP has shown that genetic programming is a viable approach to knowledge discovery, but the program is still in its infancy. The following recommendations are offered to help develop NPSGP into a mature and robust system with greater applicability to a diverse range of knowledge discovery.

1. Fitness Function Applications:

This study demonstrated that fitness function plays a critical role in determining the scope and depth of the rules generated. Comparative research is needed to determine the optimal type(s) of fitness measures to use in GP. Fitness functions should be tailored to the different data schema, e.g., quantitative, scaler, combinations of

scaler and nonscalar, etc. This would enhance GP capabilities. For example, a weighted fitness measure $((B_i + W_i) / A_i)$; where W_i represents the number of records in each category i divided by the total number of records) was successful in focusing the rules toward the smallest category in the data set, GDP GROWTH = HNEG.

Additionally, this study used a relatively simple database of quantitative data to develop and test NPSGP. Other more complex data sets may require analysis based on fitness functions using non-linear logarithmic or geometric relationships between data elements, attributes or dependent variables.

Finally, using parallel GP programs with multiple fitness measures for the same data search may expedite the knowledge discovery process.

2. Different Control Parameters

Because NPSGP is in the initial testing stage, a complete diagnosis of its capabilities was not feasible. Running GP with multiple data sets and experimenting with the GP control parameters would be very productive. At a minimum, the following control parameters should be studied to determine their potential impact on rule generation:

- a. Mutation vice Crossover, or both.
- b. Tournament size.

- c. Minimum population size as a factor of data size. The use of an extremely large population size; 100,000 and above.

3. Develop Measurements of Rule Fit

Another potential topic of research is to develop a fitness measure that indicates how the LHS values "fit" the range of data element values. For example, computing standard deviations for quantitative data and then comparing these against the range of LHS values in the rule may indicate how tight the rule value ranges fit the data set. This measure may indicate how narrow or broad the rule fits the range of attribute values, allowing another quantitative measure of the quality of the rule.

4. Database Objects

The ability to define virtual data attributes offers great potential for knowledge discovery systems. Defining new objects by combining attributes and/or specifying ranges for attribute values focuses the knowledge discovery system on multiple dependent variables packaged into a "virtual" attribute. For example, a virtual attribute "EQUIPMENT CASUALTY" may be used to define an equipment failure as a combination of attributes OPERATIONAL, INSTALLATION CODE, CRITICAL MISSION CODE, etc. If the equipment is non-operational, installed on the aircraft and critical to a mission area, you have an instance of EQUIPMENT CASUALTY.

This "virtual" attribute would help focus the knowledge discovery search on a predefined set of attribute values.

C. FUTURE DOD APPLICATIONS

The future of knowledge discovery systems lies in its ability to produce meaningful associations from data elements in existing databases. This research has demonstrated that new knowledge discovery systems involving genetic programs, such as NPSGP, offer significant potential for knowledge discovery.

As the largest single organization in the world, the Department Of Defense (DOD) has an abundance of databases that could be analyzed using the knowledge discovery techniques outlined in this thesis. The range and use of these databases are as varied as in the commercial sector. IDIS, for example, is already being used by the U.S. Army and Air Force Exchange systems. Logistic and maintenance system applications are but two additional future targets for knowledge discovery systems. The potential benefit in applying GP knowledge discovery systems to DOD databases is conceptually large, offering insights to undiscovered data relationships that may have operational and financial implications.

APPENDIX A TEST DATA

Test data used in evaluating the statistical, IDIS and NPSGP programs consisted of the following econometric data:

1. YEAR: The corresponding year for the data.
2. GDP GROWTH: As explained in chapter IV, GDP Growth was assigned a discrete value of four categories; high positive (HPOS or 0), low positive (LPOS or 1), low negative (LNEG or 2), and high negative (HNEG or 3). These categories were determined by using the growth (decline) in Indexed GDP from the previous year. Indexed GDP was taken from [Ref. 16:p. 341-474] and represents an implicit price deflator (1987 = 100). HPOS corresponds to a growth greater than 2.5 per cent. LPOS represents a growth from 0 to 2.5 per cent and LNEG represents growth from -0.001 to -2.5 per cent. HNEG growth represents any decline in GDP greater than -2.5 per cent.
3. IDXPCEPER: Indexed Personal Consumption (1987 = 100) as a percentage of GDP.
4. PCEPERCHG: The change in indexed Personal Consumption Expenditures from the previous year.
5. INDXGPDIPER: Indexed Disposal Income as a percentage of GDP.
6. GDPIPERCHG: The change in Personal Income as a per cent of GDP from the previous year.
7. EXPPERGDP: Exports as a per cent of GDP.
8. SAVPERDPI: Personal Savings as a percent of Disposal Personal Income.
9. SAVPERCHG: The change in Personal Savings from the previous year.
10. POPCHGPER: The percentage change in U.S. population from the previous year.

11. BOND10YR: The rate on 10 year U.S. Treasury Bonds, or equivalent. Years 1930 - 1932 and 1934 - 1938 were estimates.
12. CHGBDRATE: The change in the 10 year U.S. Treasury bond rate, or equivalent, from the previous year.
13. UNEMPCHG: The percentage change in civilian U.S. unemployment from the previous year.
14. UNEMPCIV: U.S. civilian unemployment.
15. DEFPEROUT: The U.S. deficit as a percentage of U.S. government outlays.
16. GDEBTGDP: U.S. government gross debt as a per cent of GDP.
17. PFDBTGDP: U.S. government public debt as a per cent of GDP.
18. GDBTPERGDP: The percentage change in U.S. government gross debt as a per cent of GDP from the previous year.

APPENDIX B IDIS PARAMETERS

IDIS provides adjustable control parameters permitting the user to focus rule search toward specified goals. These parameters are listed below with brief explanations.

1. **SAMPLING PERCENTAGE:** This parameter is used to set the proportion of the database to be used in knowledge discovery.
2. **INTEREST LEVEL:** This parameter is used to set the interest level of the independent variables. **INTEREST LEVEL** sets priorities on those LHS attributes selected in the rule search. The higher the setting the higher was the user's interest in that attribute.
3. **MAXIMUM LENGTH OF RULE:** This parameter sets the maximum number of LHS attributes that appear in the rule (i.e., the number of combine LHS attributes the generated rule should not exceed).
4. **MAXIMUM MARGIN OF ERROR:** The **MARGIN OF ERROR** calculated in rule production gives the range of estimated error of the rules computed confidence level. As a control parameter setting, **MARGIN OF ERROR** expresses the user's tolerance for error in the estimate of the confidence level. Calculation of this parameter is proprietary and has not been disclosed by IntelligenceWare, Inc., the developer of IDIS.
5. **MINIMUM RULE CONFIDENCE:** **RULE CONFIDENCE** is calculated in IDIS by dividing HIT-HIT observations by HIT-HIT and MISS-HIT observations. As a parameter **RULE CONFIDENCE** sets desired rule generality by specifying the minimum acceptable ratio of observations that fall into this calculated range.
6. **MINIMUM PER CENT OF DATABASE FOR RULE FORMATION:** This parameter specifies the minimum percentage of LHS instances, as a per cent of total instances, that must be contained in the rule. It is the number of records in the data set, as a percentage of total records, that are included in the rule.

7. MINIMUM NUMBER OF RECORDS USED FOR RULE FORMATION: This parameter is the integer equivalent of the MINIMUM PER CENT OF DATABASE FOR RULE FORMATION.
8. MAXIMUM GENERALITY: This parameter specifies the largest scalar range of the LHS attributes permissible for rule generation.

APPENDIX C NPSGP USERS GUIDE

The following lists a set of procedures that can be used to execute NPSGP. This listing contains a minimum set of user guidelines required to run the program using a UNIX workstation.

1. Load the data set either as an ASCII file. Delete all irrelevant data and insure the dependent variable is in defined in the first column. Saves the file as a text file with the name (econ.tab).
2. Convert the ASCII file to a "C" program structured file using the command:

```
perl define.pl -l5 econ.tab
```

NOTE: "-l5" sets the lag time for the data, the default value if none is entered is 5.

3. Recompile the program by typing the MAKE command.
4. NPSGP program is executed using the following commands:

- a. To start a new program:

```
gpc 1 50 none 4
```

- b. To restart a program from a failure checkpoint:

```
gpc -r 1 50 none 4
```

NOTE: "-r" specifies the restart from a checkpoint;
"50" specifies the total number of generations to run;
"4" represents the seed number.

5. Changing the default settings of population size, crossover rates, random seed, tournament selection, etc. can be done through the program module "Default.in". Changing these parameters does not require recompilation of NPSGP.

6. Changing the fitness function can be done in the program module "Fitness.C". NOTE: Changing this module REQUIRES recompilation of NPSGP.

LIST OF REFERENCES

- (1) Frawley, William J., Piatetsky-Shapiro, Gregory, and Matheus, Christopher J. , *Knowledge Discovery in Databases: An Overview*, KNOWLEDGE DISCOVERY ON DATABASES, 1991.
- (2) Parsaye, Karman , Chignell Mark ,Knoshafian Setrag and Wong Harry, *Intelligent Databases, Object-Oriented, and Deductive Hypermedia Technologies*, 1989.
- (3) Green, D. P. and Smith, S. F., *A Genetic System for Learning Models of Consumer Choice*, Proceedings of the Second International Conference on Genetic Algorithms, MIT. Cambridge, Hillsdale, N.J.:Lawrence Erlbaum, 1987.
- (4) Winkler, R. L. and Hays, W. L., *Statistics: Probability, Inference, and Decision*, Second Edition, Holt, Rinehart and Winston, 1975.
- (5) Weiss, Neil A., and Hassett, Matthew J., *Introductory Statistics*, Third Edition, 1991.
- (6) SAS Institute Inc., *SAS/STAT User's Guide, Release 6.03 Edition*, Cary, NC: SAS Institute Inc., 1988.
- (7) Messier, William F., Jr, and Hansen, James V., *Inducing Rules for Expert System Development: An Example Using Default and Bankruptcy Data*, Management Science.
- (8) Bundy, Alan, Silver, Bernard and Plummer, Dave, *An Analytical Comparison of Some Rule-Learning Programs*, Artificial Intelligence, April 1985.
- (9) Parsaye, Kamran and Chignell, Marke, *Intelligent Database Tools & Applications: Hyperinformation Access, Data Quality, Visualization, Automatic Discovery*, John Willey and Sons, Inc. 1993.
- (10) Parsaye, Karman, *What can IXL do that Statistics cannot?*, IntelligenceWare, 1990.
- (11) Koza, John R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.

(12) Liepins, G.E. and Hillard, M.R., *Genetic Algorithms: Foundations and Applications*, Annals of Operations Research, 1989.

(13) Burtka, Michael, *Genetic Algorithms*, The Stern Information Systems Review, published by the Stern School of Business, New York University, Spring 1993.

(14) Dejong, Kenneth A., *Genetic-Algorithm-Based Learning: Machine Learning Volume III*, Navy Research Laboratory, 1990.

(15) Nissen, Vollcer, *Papers on Economic and Evolution # 9303*; Evolutionary Algorithms in Management Science, July 1993.

(16) Council of Economic Advisers, *Economic Report of the President, January 1993*, Published by the United States Printing Office, Washington 1993.

INITIAL DISTRIBUTION LIST

- | | |
|--|---|
| 1. Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22304-6145 | 2 |
| 2. Library, Code 52
Naval Postgraduate School
Monterey, California 93943-5002 | 2 |
| 3. Mohammed A. Al-Mahmood
P.O.Box 5998
Manama, State of Bahrain | 8 |
| 4. Steven L. Smith
12097 Winona Drive
Lake Ridge, Virginia 22192 | 3 |
| 5. Balasubramaniam Ramesh, Code RA
Naval Postgraduate School
Monterey, California 93943-5002 | 5 |
| 6. William Gates, Code GT
Naval Postgraduate School
Monterey, California 93943-5002 | 3 |